IEEE Open Journal of
Intelligent Transportation
Systems

Received 5 May 2023; revised 24 June 2023; accepted 7 November 2023. Date of publication 29 November 2023; date of current version 8 January 2024.

Digital Object Identifier 10.1109/OJITS.2023.3336795

Characterization and Selection of WiFi Channel State Information Features for Human Activity Detection in a Smart Public Transportation System

ROYA ALIZADEH, YVON SAVARIA (Life Fellow, IEEE), AND CHAHÉ NERGUIZIAN (Member, IEEE)

Department of Electrical and Computer Engineering, École Polytechnique de Montréal, Montreal, QC H3T 1J4, Canada

CORRESPONDING AUTHOR: R. ALIZADEH (e-mail: roya.alizadeh@polymtl.ca)

This work did not involved human subjects or animals in its research.

This work was supported in part by the Grant from the Natural Sciences and Engineering Research Council of Canada.

ABSTRACT Robust methods are needed to detect how people are moving in smart public transportation systems. This paper proposes and characterizes effective means to accurately detect passengers. We analyze a public WiFi-based activity recognition (WiAR) dataset to extract human activity features from Channel State Information (CSI) data. To do so, CSI power changes caused by nearby human activity are analyzed. Our method first extracts multi-dimensional features using a Short-Time Fourier Transform (STFT) of CSI data to capture the relevant signal features. Since the environment of a transportation system changes dynamically and non-deterministically, we propose analyzing these changes with a heuristic algorithm that leverages a decision tree to automate a decision-making solution for feature selection. Principal Component Analysis (PCA) is performed before the decision tree algorithm. Reported results are compared with those obtained from the existing methods. Based on these results, we explore the effectiveness of various features such as the chirp rate, delta band power, spectral flux, and frequency of movement. This allows identifying and recommending the most effective features for the explored detection task according to observed variability, information gain, and correlation between features. The reported classification results show that using only the chirp rate estimated from CSI information as a feature, we achieve precision $= 83\%$, True Positive (TP) $= 94\%$, True Negative (TN) $= 91\%$ and F1-score $= 87\%$. Considering delta band power as an additional feature adds more information and allows getting higher performance with precision $= 100\%$, $TP = 97\%$, $TN = 95\%$ and F1-score $= 95\%$.

INDEX TERMS Feature extraction and analysis, classification, human activity recognition, channel state information (CSI), chirp rate, smart public transportation systems, principal component analysis (PCA), decision tree, feature selection.

I. INTRODUCTION

DATA analytics plays an important role in many aspects of city operations and public services [1], [2]. It can enable making transportation systems more efficient and smarter. It is a key part of our method outlined in Figure 1 that aims at enabling the creation of smart cities to improve urban mobility and to better manage transportation infrastructures. Additionally, applying machine learning algorithms to data collected from the environment of transportation systems can produce powerful means to improve the services that they provide [3]. The volume and availability of data in Intelligent Transportation Systems (ITSs) require data-driven approaches. This approach has received attention in a wide range of applications including signal recognition and estimation of how people are moving through various systems and how crowded a platform is [4]. That could be used in intelligent transportation systems, particularly those that need to be fully automated, driverless, operated, and monitored from

The review of this article was arranged by Associate Editor Xianfeng Yang.

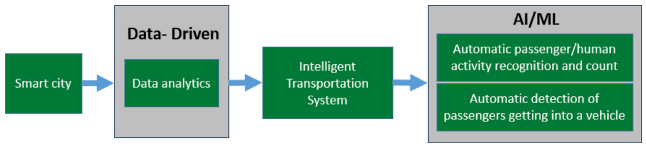

FIGURE 1. The flow of challenges in a smart city platform.

a Command Centre to ensure high levels of reliability and safety in several cities [5]. The proposed data analysis method targets non-deterministic public transportation systems that could benefit from self-learning operation in a dynamic environment [6]. Such smart data analysis platforms can use automated decision-making machine-learning-based algorithms that require IoT devices and traffic management to gather and analyze data at an unprecedented rate and to improve public transportation systems [7], [8], [9].

A data-driven centralized learning approach was proposed to predict the traffic flow to ease traffic pressure [10]. This approach is based on optimized YOLO (You Only Look Once) [11] and DeepSORT (Deep Simple Online and Real-time Tracking) [12] for vehicle detection and tracking. It was reported that the exploding vehicle traffic poses great challenges to ITSs trying to estimate transportation traffic flows. Therefore, detecting passengers provides valuable information to contribute to transportation and traffic management. ITSs have also brought great interest in developing big data based on the movement of passengers in public transportation stations. Zheng et al. [13] mentioned that characterizing the overall state of a transportation system is very challenging. Systematic approaches with different features and precisions are required to combine big data in social transportation systems. Zhu et al. [14] indicated that ITSs comprise a combination of advanced sensors, control systems, and information and communication technology (ICT). AI/ML techniques have played a vital role in estimating traffic flow in urban environments. Also, it is mentioned that decision-making-based methods are required to handle traffic flow. In light of these previously reported contributions, as Figure 1 outlines, a passenger/human activity recognition that detects those who appear to get into a vehicle is a key part of smart city platforms that could orchestrate the traveling of millions of people in cities by predicting demand and traffic [15]. It could be achievable by processing data recorded by IoT devices using advanced data analytics and AI/ML algorithms.

By contrast, detecting passengers using Infrared IoT sensors [16] as a data-driven and ML-based approach was not found to be a reliable solution. Related methods are very sensitive to noise, temperature variations, dust, and smoke. Authors in [17] detected and counted passengers in public transportation systems using cameras and computer-vision-based methods. This method is very costly, it causes a privacy issue and it does not work in dark areas. Moreover, in non-stationary environments, finding the ideal locations for cameras on the site can be difficult and may create many blind spots.

The non-computer-vision-based methods detect human mobility based on active sensing or passive sensing approaches. Active sensing methods estimate human mobility using multiple mobile sensor nodes [18]. A limitation of this method is data availability. Indeed, only the participants who have a device supporting a suitable mobile application can be detected and it requires a high training cost. The radar sensing technique reported in [19] focused on a feature-based approach to estimate human motion for real-time applications. However, the limitation of radar sensing is that it provides coarse-grain information about the presence and movement of objects. The larger-scale and fast movements could be detected by radar signals. However, stationary, or slow movements could not. Additionally, radar signals are affected by weather conditions such as rain, fog, and physical obstacles such as walls and furniture. However, WiFi-based human activity recognition provides a fine-grain, non-intrusive, and privacy-preserving approach to detecting human motions.

The present paper exploits the WiFi-based public domain WiAR dataset [20] generated from an Intel 5300 wireless NIC card. This dataset is processed with advanced data analytics and machine-learning techniques to predict human activities and to count those performing it [21]. The relationship between the velocity of movement for human activities and the frequency of the WiFi Channel State Information (CSI) data is discussed in [20]. Since the Fourier transform decomposes a signal into its constituent frequencies without providing any time information, the Short-Time Fourier Transform (STFT) is applied to capture human motion. The conventional Fourier transform produces results that are hard to interpret when dealing with time-varying signals [22]. By contrast, the STFT computes the Fourier transform over a time interval, using a sliding window that can be shifted in time and adjusted in duration.

In a transportation station, various sources of uncertainty come into play. Significant factors include humidity, echoes, environmental features and parameters, multi-path reflections, reflections from stationary and moving objects, the presence of other individuals, and interactions between two or more people. All of these factors introduce variability and stochastic (random) characteristics into the environment that make it a non-deterministic phenomenon. In the context of human activity detection, robust feature detection is essential for dealing with the challenges posed by stochastic environments because it can help to improve the accuracy and reliability of activity recognition systems. Thus, in this paper, we extract chirp rates in different frequency bands based on the duration of human activities that we wish to detect from the incoming CSI data. It was found that a chirp rate is somewhat invariant to the stochastic nature of the environment.

Chirp signals are found in many practical situations. The frequency of chirp signals either increases (up-chirp) or

decreases (down-chirp) with time [23]. They are frequently encountered in various signal processing applications, such as sonar [24], radar, and biomedical devices [25]. Some of these applications rely on chirp signal transmission as in the case of sonar [24], while others model the received signal after Doppler spread as chirp signals, e.g., in synthetic aperture radars (SARs) [26], and heart sound signals [27]. In the literature, various techniques have been developed for the estimation of chirp parameters, including the airborne SAR [28] and the Doppler frequency rate [29].

Besides considering the chirp rate as a feature in a non-deterministic environment, we take into account some other spectral and temporal STFT features to obtain the best possible performance. One of the challenges in this paper consists in determining what is the impact of correlated and non-correlated features in terms of performance. Considering correlated and non-correlated features, from a performance standpoint, there are scenarios where the algorithm might face challenges due to correlated features. Even though most correlated features can be used by the model to improve the performance, considering more features will likely slow down training and inference.

The proposed method applies the Principal Component Analysis (PCA) method and a heuristic search on the incoming CSI data to characterize features based on the highest eigenvalues. The adopted heuristic search is a kind of sequential forward search that incrementally selects the best features producing the most information gain and the least entropy using a decision tree algorithm. Performing proper feature selection using PCA allows for the reduction of complexity and the dimensionality of the data. With PCA, dimensions are created to maximize the observed variance. Moreover, it reduces the time or space required by machine learning pipelines [30]. In contrast, irrelevant features reduce the accuracy of machine learning models. In [31], adding a second feature in the training model increased the performance of the model significantly. Given the variance of a random variable, we can calculate the probability (p) associated with that variable and then compute the entropy. For example, for the Bernoulli variable X, the variance equals $p(1-p)$. And we can simply compute the entropy as $-pln(p)-(1-p)ln(1-p)$. As a result, both variance and entropy measure the uncertainty [32].

The general objective of this paper is to explore an effective tracking algorithm related to non-deterministic target dynamics. In this context, a target is a person whose movements are influenced by random or uncertain factors. The term "non-deterministic" implies that the exact trajectory or behavior of the target cannot be precisely determined in advance due to the influence of unpredictable elements. The method proposed in the present paper to analyze the observed complex CSI signals proceeds by extracting chirp parameters that correspond to specific human activities and from which these activities can be detected. This feature has never been applied to the classification of human activities. We estimate some other spectral and temporal STFT features as well. We characterize features by applying a combination of the PCA method and the decision tree algorithm to select important features to obtain the best possible performance.

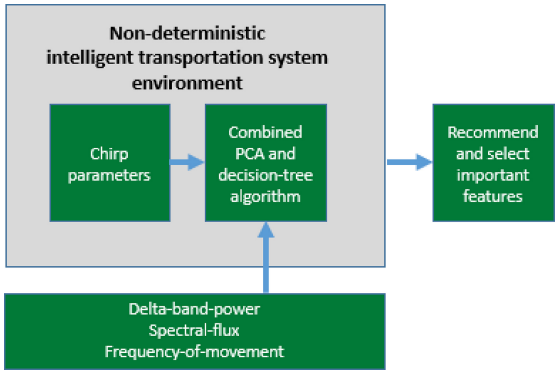

FIGURE 2. Relationship between various contributions made in the present paper.

Therefore, this paper proposes means to 1) detect people who appear to get into a smart vehicle by enhancing a processing methodology applied to the WiAR dataset, 2) extract chirp parameters to effectively detect human activities, 3) apply PCA before the decision tree algorithm to automate the feature decision-making process and to improve the model accuracy in non-deterministic environments, and finally, 4) recommend a selection of the features found to be most effective based on observed variability, information gain, and correlation between features. In summary, the contributions of this paper are illustrated in Figure 2. As this figure shows, both the estimation of the chirp rate feature and the combination of PCA and the decision tree algorithm are effective solutions for non-deterministic environments. Then, we recommend and select important features based on characterization and classification reports.

The rest of the paper is organized as follows. Section II presents the related works. The methodology and the processing framework are introduced in Section III. In Section IV, we describe the features of the STFT and how to estimate the chirp parameters for human activity from the incoming CSI data. In Sections V and VI, we present PCA and the decision tree algorithm used to select important features, respectively. In Section VII, we analyze and discuss the characterization results. In Section VIII, we benchmark the performance of the selected features by estimating the confusion matrix metrics. Additionally, we compare the results with known methods and recommend the features found to be most effective for Human Activity Recognition (HAR). In Section IX, we discuss key points regarding WiFi-based passenger motion detection. Some limitations of the present research are mentioned in Section X. Finally, conclusions and future works are drawn and proposed in Section XI.

II. RELATED WORKS

Recent research has shown the rapid development of wireless sensing and machine learning-based methods due to their

potential to enable various applications and advances in technology.

A. CROWD ESTIMATION

Authors in [33] used some preprocessing methods for WiFi-based crowd estimation by extracting 13 spectral descriptors. In total, 13 statistical features were considered: 1) mean, 2) standard deviation, 3) centroid, 4) spread factors, 5-6) 1^{st} and 2^{nd} order moments, 7) mean sigma ratio, 8) number of peaks, 9) number of valleys, 10) maximum width between peaks, 11) maximum width between valleys, and 12-13) width/height between maximum peak and minimum valley are used to extract channel information from WiFi signals reflected by the presence of human bodies. However, estimating all those statistics is a computationally expensive process. Additionally, since there is a source of uncertainty in real systems due to the stochastic nature of the environment, this technique is not as robust and reliable as extracting chirp parameters, which is found to be robust for a non-deterministic environment. WiFi passive sensing was used in [34] to count people through a wall by using deep learning methods. However, in this method, the non-deterministic characteristic of the ITS environment is not considered.

B. PASSENGER DETECTION

Delta band power and frequency of movement were extracted in [31] to detect people getting into a bus. However, as part of the work reported in the present paper, it was found that the chirp rate produces results better than the frequency of movement to achieve a robust model in a non-deterministic area. It is of interest that the frequency of movement is somehow embedded in the chirp parameters. The inspiration for extracting a chirp feature using CSI data coming from a WiFi transceiver came from the fact that chirp signals are widely used in radar systems to largely improve their range resolution and performance. Zhou et al. [35] observed the potential to apply deep learning for radar perception in autonomous driving systems. Moreover, they also proposed a deep learning system for radar perception exploiting chirp signal processing.

C. FEATURE EXTRACTION

Convolutional Neural Networks (CNNs) were used as an extraction method [36] to detect human movements from frequency-modulated continuous wave (FMCW) radar signals. These authors designed a system for spatial feature extraction from grid-like data, such as images. Although CNNs are commonly used in computer vision, they require substantial computing resources to perform motion detection on signals received from radar devices.

By contrast, Long Short Term Memory (LSTM) networks, a class of recurrent neural networks (RNNs) are well-suited for time-series data analysis [37]. They can capture long-term dependencies in the data while remembering and utilizing the context over time, thus making them suitable for tasks involving sequences and temporal patterns. Keras categorical-entropy model could be applied at the last predictive layer of the LSTM model to perform the classification of different passenger activities [21]. LSTM models are adequate for fixed and regular time intervals between sequences of elements. Classification is a challenging problem when dealing with irregularly sampled data, where the time intervals between observations vary.

Support Vector Machines (SVMs) [38] are used as robust classification methods that can be used to learn the decision boundaries based on support vectors and a margin that maximizes the separation between classes. This property makes SVMs memory-efficient and computationally effective during both the training and the inference phases. They can handle both binary and multi-class classification problems. In the method proposed in the present paper, the SVM classifier is utilized to detect passengers.

Radar devices were used in [36] to generate chirps from FMCW signals. In the present research, we were inspired by the concept of radar chirp signals and extracted the chirp rate as a robust feature that could be extracted from incoming CSI signals in a public transportation system. Since the chirp rate reflects the rate of change of a radio frequency signal rather than providing instantaneous frequency measurements, it has the potential to enhance the reliability and accuracy of the data analysis in noisy environments. In the present research, by analyzing the rate of change of frequency over time through the chirp rate, human activity can be detected reliably.

D. FEATURE SELECTION

The reported experiments were used to find the best set of features for the target application. A key point in selecting the best features is to retain those that facilitate the learning process while producing accurate activity detection. Various methods that can be used for feature selection are reviewed below.

A wrapper-based method, as proposed in [39], was employed to select features. In this method, a search procedure is defined in the space of possible feature subsets, and various subsets of features are generated and evaluated. Since the wrapper-based method goes through numerous iterations to train and evaluate different subsets of features, it is computationally expensive, especially when dealing with large datasets. Additionally, it evaluates different feature subsets to optimize the performance of the training data which may cause overfitting.

A filter-based method was proposed in [40]. In this approach, the attribute selection method is independent of the specific data-driven algorithm used for modeling and analysis. In this method, the relevance and the intrinsic characteristics are considered for feature selection. The disadvantage of this method is that it evaluates features independently of each other and does not consider potential dependencies or interactions among features. It can cause the selection of redundant or irrelevant features.

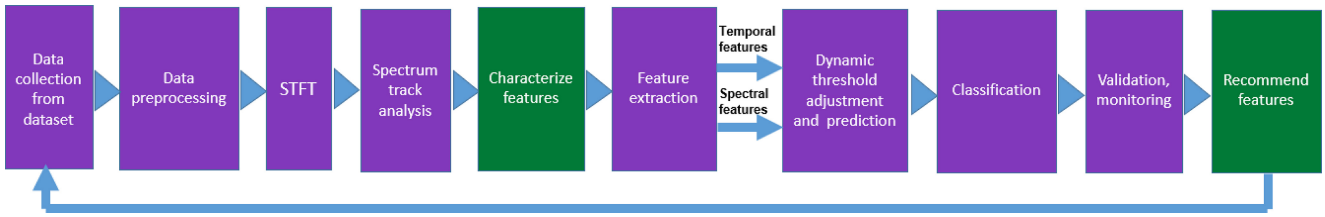

FIGURE 3. An enhanced processing framework to detect human activities.

Authors in [41] proposed a so-called relief algorithm to select features. This algorithm uses a statistical method and avoids a heuristic search. It evaluates the relevance and importance of features based on their ability to distinguish between different classes or target values. It was first proposed by Kira and Rendell [42]. However, it is sensitive to the density of the sampled data. That method estimates feature weights based on nearest neighbors. It cannot predict which features are important if the sampled data is sparse or not uniform. The relief method ranks features based on their relevance. It amplifies the weight value of features with a high correlation with the category and removes features with a low weight value. However, it cannot neglect redundant features to improve performance.

PCA [43] is widely used in machine-learning-based algorithms to reduce the number of dimensions in a set of features and to remove noise. Bharadiya [44] utilized PCA for dimensionality reduction and SVM for classification to decrease execution time by reducing the dimension of the input data. Also, it is reported that the performance of a classification algorithm was increased by this method. For instance, Wang et al. [20] used PCA to denoise CSI data. The main role of PCA is typically to reduce the dimensionality of the features set while capturing the same information.

An alternative approach for feature selection relies on decision trees [45]. In this algorithm, a decision tree is constructed by selecting features. In each layer, each feature is predicted according to some criteria such as Gini impurity [46], entropy, or information gain. This algorithm is used in our method as further described in Section VI. High classification accuracy and strong robustness are the main advantages of this method. The present paper applies the PCA before the decision tree algorithm to help automate a decision-making solution for feature selection based on the entropy and information gain to remove redundant features, reduce the dimensionality of the features, and improve the performance of the classification model.

III. METHODOLOGY

The framework proposed for HAR that is directly applicable in smart transportation systems comprises ten steps as shown in Figure 3. Its goal is to support building a more accurate model to predict human activities from WiFi CSI data. We used a public WiFi-based dataset with the data collection component of the framework [21]. The data preprocessing component encodes the features, scales the numerical features, and performs noise reduction. It is a vital pre-modeling phase for training the machine learning model successfully and preventing the overfitting problem [21]. The pre-processing module divides the dataset into two parts; the first part comprises signals captured when people appear to be waiting at a transportation station considering their standing and walking activities; the second one comprises the higher-speed (running) activities that are assumed to be analogous to people getting into a vehicle [31]. Since different people perform the same activity at different speeds and for different time durations, the STFT is performed to enable analyzing CSI signals and tracking the movement of persons in both time and frequency domains [21].

Figure 4 illustrates the spectrogram for human standing and running activities [21]. In the standing activity, the subject is walking near the end. In fact, this figure represents the energy distribution of human activities over time and frequency domains. The spectrogram is a squared magnitude of the STFT. The color bar on the right of the spectrogram indicates intensity [dB] [21]. The red color in this figure shows how the energy associated with the motion change of human activity evolves with time and frequency. Based on this figure, the most CSI energy of the running activity is concentrated at higher frequencies [21].

The goal of characterizing feature component is to choose a proper subset of features that is sufficient to describe a target. In this component, we investigate whether features allow gaining any additional information in Section VII. Moreover, before training a model to extract features, it is crucial to apply feature selection techniques and characterize them to avoid issues such as overfitting, long training time, and less reliable response to noisy input data.

The proposed framework applies PCA before the decision tree algorithm to automate the decision-making about features and their ability to transform data in directions that produce the highest variability and information gain. Features both in the temporal and spectral domains are extracted. Among the various temporal features, dominant features such as chirp rate and frequency of movement are predicted. Also, the spectral features such as delta band power, and spectral flux are considered. After deriving the values of features both in time and spectral domains, we estimate the dynamic threshold as a decision boundary of the classifier to predict specific actions such as potentially getting into a vehicle with a high accuracy [31]. We apply a 5-fold cross-validation method and confusion matrix to validate

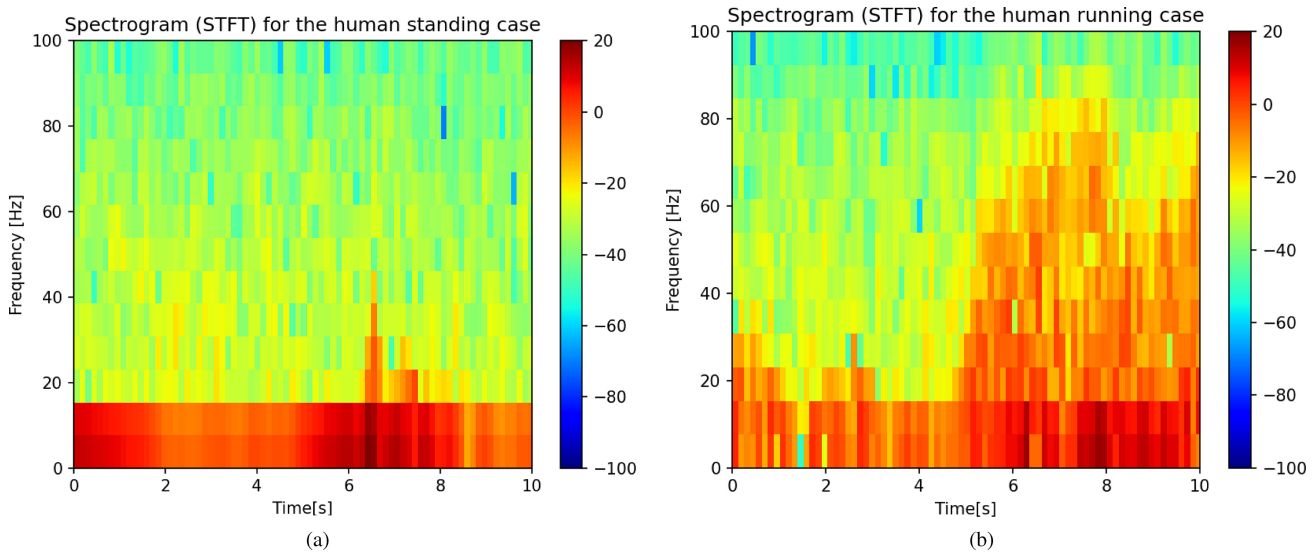

FIGURE 4. Spectrogram for two known human activities labeled a) standing b) running [21].

the model and monitor the data to predict specific actions in Section VIII after selecting the most important features. There is a trade-off between latency, accuracy, and selecting a set of features. Thus, we recommend feature selection based on observed variability, information gain, and correlation between features.

IV. STFT TEMPORAL AND SPECTRAL FEATURES

In this section, we describe temporal and spectral features related to the STFT which are important to estimate and detect the desired characteristics of the signals. The order in which features are introduced is based on their importance according to our analysis results.

Feature 1: $\mathcal{F}(1)$ = Chirp parameters - In this section, we describe what is a chirp signal, how to estimate the chirp parameters for each human activity from the incoming CSI data. A chirp signal is a signal whose frequency increases or decreases with time. Besides information on time and frequency, a chirp transform contains two more features, the chirp rate and the time window which makes it a feature with four parameters. Moreover, the chirp transform provides better localization in both time and frequency domains in the case of transient signals [47]. Mathematically, a linear chirp signal can be represented by Equation (1).

$$y(t) = exp\left[j\phi_0 + 2\pi j\left(\alpha t + \frac{\beta}{2}t^2\right)\right] \quad 0 < t < T \tag{1}$$

where α is the starting frequency, β is the chirp rate, ϕ_0 is the initial phase, T is the duration of the chirp signal [48] and j is the imaginary number, with $j^2 = -1$. To estimate the chirp parameters, we compute the STFT of the incoming CSI data for each human activity. Mathematically, the N-point STFT of a signal at time m can be expressed as Equation (2) [49].

$$X_{n,m} = \sum_{i=0}^{N-1} x_i w_{i-m} e^{-j2\pi n \frac{i}{N}}, \qquad n = 0, 1, \ldots, N-1 \tag{2}$$

where j is the imaginary number, with $j^2 = -1$, x_i is the i^{th} sample of incoming CSI data, n and m represent the corresponding frequency index and data block index, respectively. N is the number of window samples and the window sequence w_m is non-zero in the interval [0, N–1]. For each human activity, the indices of the start frequency and the end frequency are considered as n_1 and n_2, respectively. $N = n_2 - n_1 + 1$ is the number of frequency indices that cover the frequency band representative of each human activity. The spectrum matrix X consists of complex values X_{nm} as expressed by Equation (3).

$$X = \left[X_{n,m}\right],\ n = n_1, n_1+1, \ldots, n_2;\ m = 0, 1, \ldots, M-1 \tag{3}$$

where M is the number of samples in each time block. The magnitude squared of $X_{n,m}$ creates the STFT frequency time-space matrix that can be expressed as Equation (4).

$$S = \left[|X_{n,m}|^2\right],\ n = n_1, n_1+1, \ldots, n_2;\ m = 0, 1, \ldots, M-1 \tag{4}$$

The chirp parameters are then extracted from the STFT frequency time-space matrix $\mathbf{S}$ as explained below. As the chirp parameters depend on the duration and the dynamic speed of each activity, they can be used to detect human activity characteristics.

For a human activity with a time-varying frequency of movement, a chirp rate would reflect the slope of the frequency variation divided by the fundamental frequency.

$$\beta = \frac{\Delta f}{f_0} = \frac{f_{n2} - f_{n1}}{f_0} \tag{5}$$

Algorithm 1 Estimate Chirp Parameters

1: Collect CSI data for each human activity
2: Apply the STFT transform
3: Estimate chip rate β from Equation (5)
4: Estimate α, T from the STFT of the incoming CSI data
5: Estimate ϕ_0
6: Calculate Equation (1) by estimated parameters $(\alpha, \beta, \phi_0, T)$

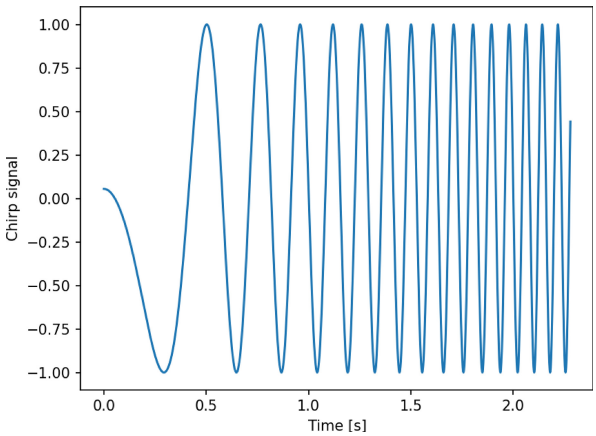

FIGURE 5. Chirp signal produced using Python from parameters extracted from STFT of CSI data for a standing activity. The standing activity includes a walking activity near the end.

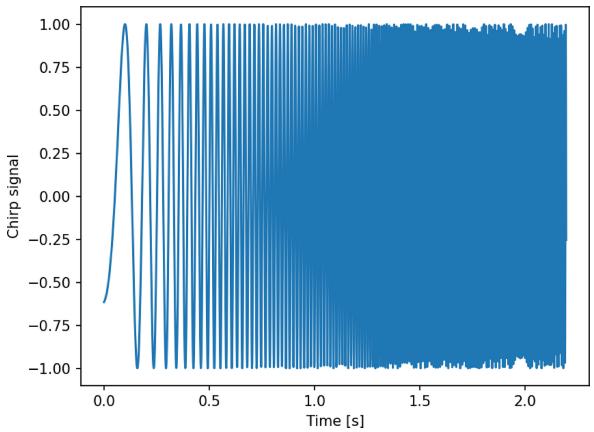

FIGURE 6. Chirp signal produced using Python from parameters extracted from STFT of CSI data for a running activity.

Algorithm 2 Delta Band Power [31]

1: Divide the dataset into segments
2: Estimate PSD for each segment
3: Compute the band power using the PSD estimate in a specific frequency range $[f1, f2]$
4: Calculate the change of band power in the frequency range of $[f1 + \Delta f, f2 + \Delta f]$ and $[f1, f2]$

where f_0 is the center frequency of the signal produced by the 5300 NIC card for 5 GHz band that is equal to 5.825 GHz and Δf is the difference between the maximum and minimum frequency for each human activity. We calculate T from the time range during which significant energy is observed according to the STFT energy threshold ξ expressed as Equation (6). This time range T becomes the estimated duration of each activity.

$$T = max\{\Delta t | \frac{\Delta Energy}{TotalEnergySTFT} \geq \xi\} \tag{6}$$

We consider $\alpha = f_{n1}$ for each human activity as the starting frequency. We calculate the initial phase from a spectral analysis using the Hilbert transform [50]. Algorithm 1 summarizes the method to estimate the chirp parameters of chip rate β, starting frequency α, duration of the activity T, and initial phase ϕ_0. Substituting the obtained parameters in Equation (1), a chirp signal can be synthetically produced using Python. Figures 5 and 6 show the chirp signals of standing and running activities, respectively. The standing activity includes a walking activity near the end.

Feature 2: $\mathcal{F}(2)$ = *Delta Band Power* - Algorithm 2 describes how to estimate the delta band power. It is derived from the Power Spectral Density (PSD), which facilitates the calculation of the cumulative power in the spectrum of a signal. We proposed the concept of the delta band power in [31] with the method to estimate it.

The PSD is estimated from chunks of input data. The length of each segment is based on the frequency resolution required for the PSD analysis. It is typically a power of two for efficient Fourier transform calculations. The frequency resolution is defined as:

$$F_{res} = \frac{F_s}{N} = \frac{F_s}{F_s \tau} = \frac{1}{\tau} \tag{7}$$

where F_s is the sampling frequency of the signal, N is the total number of samples and τ is the duration in seconds [31]. In the present research, we considered $NFFT = 128$, $F_s = 1000$, where $NFFT$ determines how many FFT points are computed in each segment.

Feature 3: $\mathcal{F}(3)$ = *Spectral Flux* - The spectral flux is computed as the difference of the squared values between the normalized magnitudes of the spectra of two successive frames. More precisely, it is calculated as the L2-norm, known as the Euclidean distance, between the two normalized spectra [51].

$$SF_p = \sum_{k=1}^{N/2} \left(|X_p[k]| - |X_{p-1}[k]|\right)^2 \tag{8}$$

where p is the frame index, k is the frequency index for the discrete Fourier transform, $X_p[k]$ and $X_{p-1}[k]$ are the normalized magnitude of the spectrum of the present and previous frames, respectively. In fact, the spectral flux indicates the change in the power spectrum of the signal between successive frames.

Feature 4: $\mathcal{F}(4)$ = *Frequency of movement* - A multipath reflection model is used to estimate the frequency of movement. The received signal from the WiFi transceiver consists of both static and dynamic paths. Therefore, the channel frequency response (CFR) $H(f, t)$ depends on both

static reflected paths such as walls and paths dynamically reflected by human activities. $H(f,t)$ is represented by [20]:

$$H(f,t) = e^{-j2\pi \triangle ft}\left(H_s(f) + \sum_{k=1}^{P_d} a_k(f,t)e^{-j2\pi f \tau_k(t)}\right) \quad (9)$$

where H_s is the static CFR, $a_k(f,t)$ and $\tau_k(t)$ are the complex channel attenuation and the time of flight for path k, respectively, and P_d is the existing set of dynamic paths. The carrier frequency offset (CFO) $e^{-j2\pi \triangle ft}$ is a phase shift caused by the carrier frequency difference $\triangle f$ between the sender and the receiver [52]. The signal propagation distance $d_k(t)$ is:

$$d_k(t) = c\tau_k = f\lambda\tau_k \quad (10)$$

$$f\tau_k(t) = d_k(t)/\lambda = \frac{\int_{-\infty}^{t} \upsilon_k(u)du}{\lambda} \quad (11)$$

where λ is the wavelength, c is the speed of light, f is the frequency of the path length change, τ_k is the time of flight for path k, and $\upsilon_k(u)$ is the rate of path length change. The frequency of path length change refers to the rate at which the path length between the transmitter and the receiver changes due to reflections caused by some human movement. The CFR power is obtained by $|H(f,t)|^2$. H_d is a dynamic CFR and is given by:

$$H_d(f,t) = \sum_{k\in P_d} a_k(f,t)e^{-j2\pi d_k(t)/\lambda} \quad (12)$$

Equations (11) and (12) relate the variation of CSI power to the movement speed [20]. It demonstrates that the CFR power depends on the rate of path length change which is caused by human activities. For example, the running activity has the highest rate of path length change in higher frequencies, while the walking activity has a lower rate of path length change in lower frequencies. However, standing activity has no path length change. As a result, the rate of path length change $\upsilon_k(u)$ is related to the frequency of movement.

Based on Parseval's theorem the total energy of a signal in the time domain equals the spectral power in the frequency domain [53]. We estimate the energy/power in the frequency domain over a time interval by transforming the STFT of the human activity CSI signal. The frequency range obtained by the STFT transform reflects the frequency of movement.

V. PCA

PCA is used for dimension reduction of large datasets, by transforming a large set of variables into a smaller one that still contains most of the information and minimizes the projection error. It creates a set of linearly uncorrelated axes called principal components (PCs) in a way that the first principal component captures the maximum variance in the data, the second principal component captures the second largest variance (while being orthogonal to the first), and so on [54]. An important question with PCA is finding how many dimensions we need to analyze the data. The first metric to answer the question is the redundancy in the dataset. Ideally, the basis vectors of the PCA should not create redundant data. Redundant data refers to data features that do not contribute significantly to the overall variance in the dataset. Redundant features are highly correlated with one another and with other features in the dataset, meaning that they carry essentially the same information as other variables.

FIGURE 7. A range of potential redundancies within a dataset a) Uncorrelated dimensions, b) Partly correlated dimensions, and c) Strongly correlated dimensions.

Algorithm 3 PCA Algorithm [43]

1: Normalize input data
2: Calculate the covariance matrix of input data
3: Calculate eigenvectors and eigenvalues
4: Sort the eigenvectors according to eigenvalues in a decreasing order
5: Choose first k eigenvectors that will be the new k dimensions

Figure 7 illustrates a range of potential redundancies within a dataset. The data in Figure 7 (a) are uncorrelated because one cannot predict one from the other. Conversely, the data in Figure 7 (c) are highly correlated, indicating highly redundant data. The data in Figure 7 (b) are partially correlated. Removing redundant and correlated features reduces the dataset dimensionality. The second metric is the covariance measure. A large positive value of covariance shows a high correlation between two features. A large negative value represents that two features are negatively correlated. It describes when two variables tend to move in opposite sizes and directions from one another, such that when one increases the other variable decreases, and vice-versa. The covariance value for uncorrelated features is zero. If we consider features as a column vector of H, where h_i represents the i^{th} feature, the covariance matrix C_H is represented by Equation (13).

$$C_H = \frac{1}{N}HH^T \quad (13)$$

Eigenvectors of covariance matrix $H \times H^T$ with dimension of $N \times N$ are calculated, where N is the number of samples. The diagonal elements of C_H represent the variance of feature vectors, where large values correspond to significant features. The non-diagonal terms denote the covariance between features, and large values indicate high redundancy in the data.

Algorithm 3 summarizes how to perform the PCA. Computing the eigenvectors and sorting them in descending order of eigenvalues allows us to create the principal

components in order of significance. Discarding components with lesser eigenvalues, we can generate a matrix of feature vectors. The columns of this matrix are the eigenvectors of the principal components. Moreover, the i^{th} diagonal value of C_H is the variance of H along h_i or feature i.

The eigenvectors of the covariance matrix are the directions of the axes where there is the most variance. They are called the principal components. Eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each principal component. The features with the largest variance value present the most important features of the data and they are associated with the highest signal-to-noise ratio (SNR). Features in the resulting ordered set are orthogonal to each other when the covariance matrix is a diagonal matrix.

Mathematically, PCA is computed using the singular value decomposition (SVD) expressed by Equation (14) [55].

$$H = U\Sigma V^T \tag{14}$$

where $V = \{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_m\}$ are orthonormal eigenvectors with corresponding eigenvalues λ for symmetric matrices $H^T H$, with $(H^T H)\hat{v}_i = \lambda_i \hat{v}_i$, $\sigma_i \equiv \sqrt{\lambda_i}$. The diagonal matrix Σ is the rank-ordered set of singular values σ_i, and U is an $m \times m$ unitary matrix, while $\hat{u}_i \equiv \frac{1}{\sigma_i} X\hat{v}_i$, $XV = U\Sigma$ [55]. Based on Equation (13), the principal components of H are the eigenvectors of HH^T. Then, from the covariance matrix C_H, we can calculate the Pearson correlation coefficient (PCC) matrix R using Equation (15) [56].

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} C_{jj}}} \tag{15}$$

PCC is used to determine the degree of correlation between features. Meanwhile, PCA is utilized to quantify the relative importance of each dimension by measuring the variance along each principal component. In fact, we can project our original dataset on the orthogonal directions identified as principal components and for which data has significant variance along them.

A related concept is the explained variance, which is a statistical measure characterizing how much variation in a dataset is related to each principal component (the eigenvector) generated by the PCA method. Specifically, it shows the total variance "explained" by each component which allows ranking components following an order of importance [57]. The importance of each principal component is reflected by the magnitude of the corresponding values in the eigenvectors. A higher score of explained variance means that the component is more significant. The explained variance for each eigenvector (principal component) $\hat{v}_i$ is expressed as the ratio of eigenvalue λ_i and the sum of all eigenvalues as Equation (16). The sum of all explained variance equals one and the explained variance for partly redundant features is small.

$$Explained\ variance(\lambda_i) = \frac{\lambda_i}{\lambda_1 + \lambda_2 + \cdots + \lambda_m} \tag{16}$$

Algorithm 4 Steps in the Iterative Dichotomiser 3 Algorithm [45]

1: It begins with the original set S (our dataset) as the root node
2: It iterates through each feature of the set S and calculates entropy ($\mathscr{H}$) and information gain (I) of features
3: It selects the feature that has the smallest entropy or the largest information gain
4: The set S is then split by the selected feature to produce a subset of data
5: The algorithm recurs on each subset

To answer the question of how many principal components should be considered, we characterized it with performance metrics presented in Sections VII and VIII. Since PCA transforms the data in the directions that have the highest variance, it inherits the property of the highest information gain. In the next section, we describe the decision tree algorithm used to visualize features based on the highest information gain.

VI. DECISION TREE AND ENTROPY

In this section, we describe how to use entropy and information gain to build a decision tree to find the most relevant features to achieve better performance. In a decision tree, all data is considered as a root node. One feature is chosen and is split into two groups. Then another feature is chosen to split data further. The recursive splitting continues until a stopping criterion is met. This criterion could be a predefined maximum depth of the tree, a minimum number of samples required to split a node, or the remaining error of the resulting subsets falling below a threshold. A suitable stopping criterion prevents overfitting and ensures that the tree does not become too complex. In this paper, the stopping criterion is defined as the maximum depth of the tree. The last node is called the terminal/leaf node. The nodes between the root node and the leaf node are called decision nodes. The decision node contains a condition to split the data. The condition is defined by the feature index and the threshold value for that feature. The leaf node helps to predict the class of a new data point. An important question is to determine which feature to use for splitting the data. Algorithm 4 describes the Iterative Dichotomiser 3 (ID3) method that we used to build a decision tree and to select features that produce high information gain [45]. This algorithm learns which features and the corresponding threshold values to use to optimally split the data. The model selects the split that maximizes the information gain. Once the tree-building process is complete, the samples end up in the leaf nodes. For classification tasks, the majority class of samples in a leaf node becomes the predicted class for any new sample that falls into that node [45].

The main advantage of the decision tree algorithm is that it uses different feature subsets and decision rules at different stages. It can capture descriptive decision-making knowledge from training sets. The goal of machine learning models is to use entropy to reduce uncertainty to build a decision tree.

Shannon defined the entropy $\mathscr{H}$ of a discrete random variable X when it takes values from an alphabet χ that is distributed according to $p : \chi \rightarrow [0, 1]$ [58].

$$\mathscr{H}(X) = E[I(X)] = E\big[-logb\ p(X)\big] \tag{17}$$

where E is the expected value operator and I is the information content of X that is a random variable. The entropy can be expressed as:

$$\mathscr{H}(X) = -\sum_{x\epsilon\chi} p(x) log_b\ p(x) \tag{18}$$

where b is the base of the logarithm used. The variance of a random variable X is defined by Equation (19) [59].

$$Var(X) = E\Big[X^2\Big] - E[X]^2 \tag{19}$$

Based on the definition of variance as Equation (19), estimating the expected value of the square of the random variable leads to estimating its variance. For binary data classification, entropy is presented by Equation (20) [45].

$$\begin{aligned}\mathscr{H}(S) &= -\sum_{i=1}^{c} p_i\ log_2(p_i) \\ &= -p_1\ log_2(p_1) - p_2\ log_2(p_2)\end{aligned} \tag{20}$$

where p_i is the probability of class i in our dataset S and c is the number of classes which is 2 in our case, $p_1 = \frac{N_{PeopleWaitForVehicle}}{N_{Total}}$ and $p_2 = \frac{N_{PeopleGetIntoVehicle}}{N_{Total}}$. The first class is related to people at a transportation station waiting for a vehicle, and the second class is related to people getting into a vehicle. The information gain is the reduction in entropy obtained by transforming a dataset and it is often used in training decision trees. It is calculated by comparing the entropy of the dataset before and after a transformation and it is the basic criterion used to decide whether a feature should be used to split a node or not [60]. It is also called mutual information $I(Y; X)$ in information theory.

$$I(Y; X) = \mathscr{H}(Y) - \mathscr{H}(Y|X) \tag{21}$$

As Equation (21) describes, we simply subtract the entropy of Y given X from the entropy of just Y to calculate the reduction of uncertainty about Y given an additional information X about Y [61]. In our case, when we build a decision tree, we want to pick up the features that will give us the greatest reduction of uncertainty. This decision tree is the one offering the largest information gain.

VII. CHARACTERIZATION RESULTS

Using PCA to reduce dimensionality and computational complexity, we need to decide which features and how many features to use. Table 1 illustrates the most important features in decreasing order based on the explained percentage of contributed variance. This table shows that the chirp rate is the most important feature among the four features considered in the present paper (chirp rate, delta band power, spectral flux, and frequency of movement).

TABLE 1. The most important features using PCA.

PC1	chirp-rate
PC2	delta-band-power
PC3	spectral-flux
PC4	frequency-of-movement

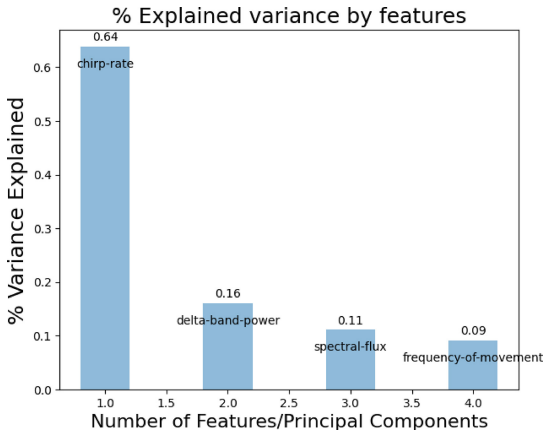

FIGURE 8. Explained variance of the considered features.

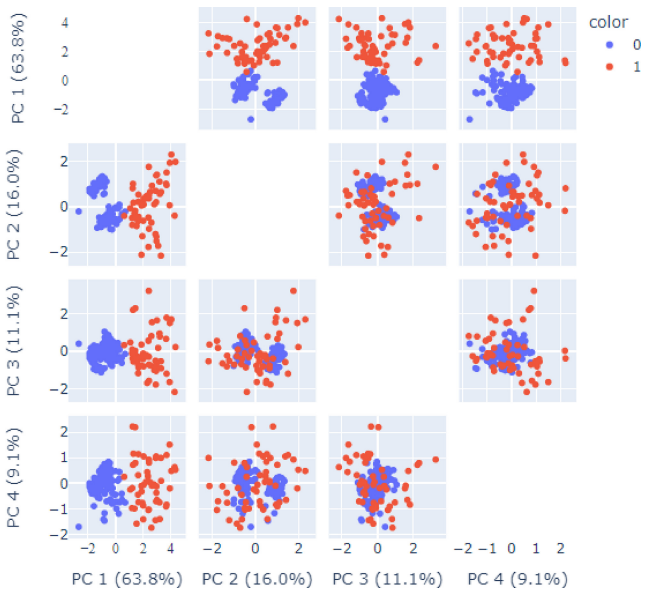

FIGURE 9. PCA visualization for the four retained features: chirp rate, delta band power, spectral flux, and frequency of movement.

We use "Scree/Elbow plot" [62] to decide how many features to use. Using this method, we plot the percentage of explained variance versus the number of features. As Figure 8 shows, the first feature explains over 64% of the variance. And the next ones explain a smaller percentage of the variance. This graph shows that at the 4^{th} feature, the variance decays to an almost insignificant level, less than 0.1 maximum variance. As a result, we use the first three features to analyze the data. The total variance explained using these three components is 91%. The corresponding PCA explained variance ratio for those three features is: [0.64, 0.16, 0.11]. PC1 explains 64% of variability, PC2 16% and PC3 11%.

TABLE 2. Pearson correlation coefficients.

	Chirp rate	Delta band power	Spectral flux	f movement
Chirp rate	1	0.54	0.538	0.598
Delta band power	0.54	1	0.411	0.418
Spectral flux	0.538	0.411	1	0.57
f movement	0.598	0.418	0.57	1

Another way to determine how many features to choose is to define a threshold like 85% or 95% that the PCA explained-variance-ratio is required to meet. Figure 9 illustrates PCA visualization for all four features. In this figure, blue and red colors correspond to the class 0 and 1, respectively. It illustrates the variance of each principal component using the covariance matrix and eigenvectors and ranks them by their relevance (explained variance/eigenvalues). Moreover, it shows the features required to easily separate two different classes. It shows that PC1 contains the highest explained variance and is the most important feature in separating two classes. The combination of PC1 with other principal components adds more information and increases the performance scores discussed in Section VIII.

Table 2 presents the Pearson correlation coefficients for the four considered features: 1- chirp rate, 2- delta band power, 3- spectral flux, and 4- frequency of movement. The Pearson correlation coefficient measures the strength and direction of the linear relationship between two quantitative variables. Specifically, we can test if there is a significant relationship between the two features. If the non-diagonal elements are zero, it indicates that those features are orthogonal. This table shows that the features are partially orthogonal. It is found that the most significant linear relationship exists between the chirp rate and the frequency of movement. Indeed, the frequency of movement is a subset of the chirp rate and it is a partly redundant feature. We use the covariance matrix when the variability of the data is significant, while we can use the correlation matrix when the angle of the basis function and the orthogonality are important.

The Python Scikit-learn package [63] is used to build and optimize the decision tree classifier. Figure 10 illustrates the ID3 algorithm used to build a decision tree before applying the PCA. In each node, an entropy greater than zero implies that samples contained within that node belong to different classes. An entropy value of zero means that the node is pure and only a single class of samples exists. The sample value in each node indicates the number of samples in the corresponding feature. The value list represents how many samples at the given node fall into each category. The first element of the list shows the number of samples that belong to the class 0, and the second element of the list shows the number of samples that belong to the class 1. The class value shows the prediction of a given node and it is determined from the value list. This method uses color to indicate the majority of the class and the extremity of values for splitting and creating multiple output branches. The leaf node indicates the predicted value. This value is the majority

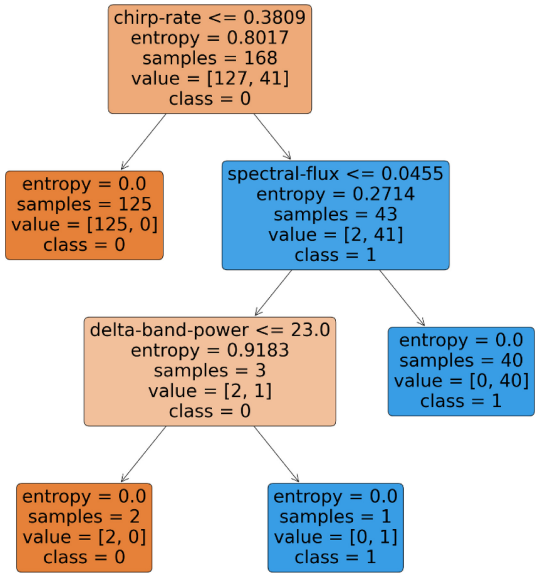

FIGURE 10. Decision tree algorithm based on entropy for features of chirp rate, delta band power, spectral flux and frequency of movement before applying PCA.

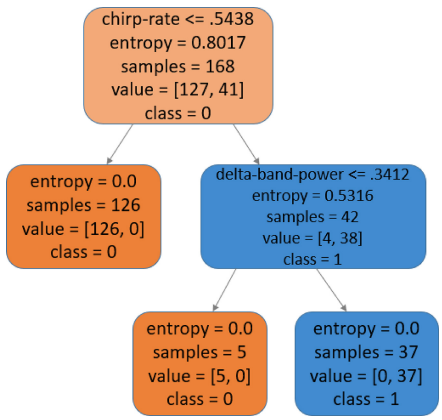

FIGURE 11. Decision tree algorithm based on entropy for features of chirp rate, delta band power, spectral flux, and frequency of movement after applying the PCA.

class of the leaf node. It determines the class of a new data point if the data point ends up in this particular leaf node [64].

The Python Scikit-learn package provides a decision tree classifier implementation that calculates the entropy and the information gain of the whole dataset and the individual feature values in each decision node using Equations (20) and (21). The feature that contributes the highest information gain becomes the splitting criterion for the root node, and the samples are divided into child nodes based on their values. The same process is repeated for each child node. Thus, at each node, the remaining features are evaluated by the algorithm and the most important feature for further partitioning is selected. This process continues until a certain condition is met, such as reaching a maximum depth or achieving a minimum number of samples per leaf node.

As mentioned in Figure 10, the decision-tree-classifier function of the Scikit-learn package is based on the ID3 algorithm that gives the criteria defined to be to chirp-rate

TABLE 3. SVM classifier confusion matrix for the training set and the test set.

		chirp-rate								chirp-rate and delta-band-power							
		Training set, Norm=None		Training set, Norm=True		Test set, Norm=None		Test set, Norm=True		Training set, Norm=None		Training set, Norm=True		Test set, Norm=None		Test set, Norm=True	
True label	0	TP:118	FN:6	TP:0.95	FN:0.048	TP:30	FN:2	TP:0.94	FN:0.062	TP:122	FN:0	TP:1	FN:0	TP:32	FN:1	TP:0.97	FN:0.03
	1	FP:12	TN:32	FP:0.27	TN:0.73	FP:1	TN:10	FP:0.091	TN:0.91	FP:7	TN:39	FP:0.15	TN:0.85	FP:0	TN:10	FP:0	TN:1
		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		Predicted label				Predicted label				Predicted label				Predicted label			

TABLE 4. SVM classifier confusion matrix considering important features for the training set and the test set.

		chirp-rate, delta-band-power and spectral-flux							
		Training set, Norm=None		Training set, Norm=True		Test set, Norm=None		Test set, Norm=True	
True label	0	TP:127	FN:1	TP:0.99	FN:0.0078	TP:32	FN:0	TP:1	FN:0
	1	FP:1	TN:43	FP:0.03	TN:0.97	FP:1	TN:10	FP:0.046	TN:0.95
		0	1	0	1	0	1	0	1
		Predicted label				Predicted label			

≤ 0.3809 to split data for the root node. The entropy that corresponds to this criterion is equal to 0.802. The left-most leaf node is created with this condition. Now the algorithm should split data from the right branch with the condition of spectral-flux ≤ 0.0455 to the left and to the right. In this way, the right child is a pure leaf node. The algorithm splits data further with those conditions to build the decision tree. Finally, delta band power with a condition of delta-band-power ≤ 23.0 creates two pure leaf nodes that do not require more split. In this figure, the depth of a tree is three.

Figure 11 illustrates the results of the ID3 algorithm used to build a decision tree after applying PCA to those four features. Applying PCA before the decision tree algorithm allows effective feature characterization. PCA is used to select features based on the highest eigenvalues, which leads to minimizing the total number of features and amplify the classification performance by transforming data to the directions that have the highest variability. Additionally, it makes the decision tree learn faster and achieve a higher accuracy with the least number of features. Since the decision tree is applied on the aligned data with the highest variability, after splitting the data with a condition of chirp-rate ≤ 0.544, delta band power creates the most information gain and two pure leaf nodes. Therefore, it reduces the complexity of computation from three features to two.

VIII. SELECTION OF THE MOST IMPORTANT FEATURES

To ensure that we choose the right number of features, we cross-validate the performance of classification providing the confusion matrix as Tables 3, 4, and 5. The computation and prediction are performed using Intel Xeon *X*86 processor. Python libraries of Scipy, Numpy, and Sklearn are used for estimating and characterizing important features. Among the total 3400 available samples of different human activities using the WiAR dataset, we estimated running, standing and walking to detect people who appear to get into a vehicle. The features of chirp rate, delta band power, spectral flux, and frequency of movement with different combinations are used as the input to the SVM classifier based on supervised machine learning. A 5-fold cross-validation approach was applied, where 80% of the data is used as a training set and 20% is used as a test set.

Tables 3, and 4 report the SVM classifier confusion matrix for three use cases of 1) chirp rate, 2) chirp rate and delta band power), and 3) chirp rate, delta band power and spectral flux. Norm = True presents the normalized confusion matrix in the range of [0, 1]. The rows of the table show the true classes of the people waiting for a vehicle and people who appear to get into it, whereas the columns show the predicted classes as estimated by the SVM linear-kernel classifier. An ideal result is to obtain near 100% classification on the diagonal of the confusion matrices, whereas near zero on the non-diagonal elements related to misclassification results.

Table 5 illustrates the performance of the SVM model in terms of the classification metrics for the test set. The performance metrics include accuracy, precision, recall, and F1-score based on the following equations [65].

$$Accuracy = (TP + TN)/(TP + FP + TN + FN) \quad (22)$$

$$Precision = TP/(TP + FP) \quad (23)$$

$$Recall = TP/(TP + FN) \quad (24)$$

$$F1_{score} = 2\frac{Precision \times Recall}{Precision + Recall} \quad (25)$$

Confusion matrix metrics of *TP*, *TN*, *FP*, and *FN* are true positive, true negative, false positive and false negative, respectively [31]. Equation (22) represents the accuracy of the model. It is the ratio of TP and TN to all positive and negative observations. Equation (23) describes the precision metric which measures the proportion of positively predicted labels that are correct. Equation (24) represents the recall metric which measures the correctly predicting the positives out of actual positives. This equation shows that FN impacts the recall score. The lower FN results in the higher recall score. The recall score, also known as the true positive rate or sensitivity, primarily measures a model's ability to correctly identify positive samples while potentially sacrificing the accuracy of negative sample identification. It focuses on minimizing false negatives. Equation (25) represents F1-score which is the harmonic mean of precision and recall.

According to the results in Tables 3 and 5, using only chirp rate as a feature, we achieve $TP = 94\%$, $TN = 91\%$, accuracy $= 93\%$, precision $= 83\%$ and F1-score $= 87\%$. When we add the second important feature of delta band power, it improves the performance to $TP = 97\%$, $TN = 100\%$, accuracy $= 98\%$, precision $= 100\%$, and F1-score $= 95\%$ by giving more information gain. Based on Tables 4 and 5, when we consider all three important features of chirp rate, delta band power and spectral flux, the information gain increases more and it improves the performance to $TP = 100\%$, F1-score $= 98\%$ to detect

TABLE 5. Classification metrics.

	Test set					Training set
	chirp-rate	chirp-rate, △band-power	△band-power, f-mov	chirp-rate, △band-power, spectral-flux	all-4-features	all-4-features
Accuracy	0.93	0.98	0.98	0.98	0.72	0.98
Precision	0.83	1	0.91	1	0.64	1
Recall	0.91	0.9	1	0.91	0.69	0.97
F1-score	0.87	0.95	0.95	0.98	0.66	0.98

people getting into a vehicle. Therefore, we obtain the most information gain, the highest performance and scores by adding a second and a third important features.

To compare the proposed method with the results reported in [31], Table 5 shows that the classification metrics and performance of delta band power and frequency of movement are roughly as high as the ones for chirp rate and delta band power. However, based on Figure 8 and Table 2, frequency of movement results in the lowest explained variance and the most correlation with chirp rate feature. The reason is that the frequency of movement is a subset of the chirp rate. The higher chirp rate corresponds to the higher frequency of movement, while the lower chirp rate captures the lower frequency of movement for effective motion detection. Additional parameters such as duration of activity, rate of frequency changes, minimum and maximum frequency of movement are extracted and estimated using chirp rate as a feature which makes the model robust and adequate for non-deterministic environments. The duration of the activity is important to distinguish people who are waiting at a transportation station and the ones around there. Moreover, the variable duration of the chirp feature achieves the energy-efficient system design. The lesser selection of features makes the lesser consumption of resources and computational time.

Based on Table 5, considering all four features cause the overfitting problem, higher computational complexity, and higher latency. Overfitting occurs when a model performance on the training dataset is improved at the cost of worse performance on data not seen during the training, such as a test set or new data. This table shows that using all four features for the training set we achieve high performance while for the test set, we obtain F1-score $=$ 66% and precision $=$ 64%. In fact, it is the place where the algorithm breaks down.

Characterizing features by applying PCA before the decision tree algorithm allows the selection of the most important features based on the highest eigenvalues and information gain. Evaluating classification metrics helps us understand the strengths and limitations of our model when making predictions in new situations. Based on the results obtained in this section, the chirp rate contributes the highest variance and information gain among the other features.

In terms of classification metrics, there is a trade-off between false positives and false negatives. That is why recall and precision have to be considered together. If the goal is to reduce false negatives, a high-precision model should be used. By contrast, if the goal is to decrease false positives, a model with a high recall should be employed. For example, if a model with a high accuracy achieves a precision equal to 50%, it means that 50% of the time, when the model predicts a certain activity, it is not actually the case. Moreover, a low recall score specifies that the model is not good at identifying *TP* and *FN* examples. Based on Table 5, if precision $=$ 83% and F1-score $=$ 87% are considered, estimating chirp rate as a feature meets the requirements. If high precision is required, we should select chirp rate and delta band power features. If high recall is required, delta band power and frequency of movement are proper choices. If the highest F1-score is required, we should select all three important features.

IX. DISCUSSION

In this paper, a data processing framework was proposed to detect passengers who appear to get into a vehicle of a public transportation station using the public domain WiAR dataset based on *IEEE* 802.11*n* CSI measurements. Additionally, a practical approach was proposed to characterize and select features for passenger activity detection. The following summarizes the key points to consider with respect to WiFi-based human motion detection:

- Identifying congestion areas: In WiFi-based smart public transportation systems subject to congestion, this issue could be addressed by: 1) channel selection and configuration to improve performance, 2) load balancing to distribute the traffic evenly across multiple access points to prevent congestion 3) traffic monitoring and analysis to identify areas of congestion and to optimize network resources.
- Privacy protection: In this research, the proposed detection method is based on reflected signals emitted by the WiFi system. It is not related to a device that belongs to a user such as a cell phone. It only detects the presence and the movement of passengers in the environment. Therefore, it can hardly introduce any privacy issues.
- Real-world application scenario: Validation of the proposed concept with a real-world transportation system would be very interesting and highly valuable. However, the present work is based on and was enabled by the public domain WiAR dataset, and thus validation in a real-world transportation system is beyond its scope.

X. LIMITATION OF THIS RESEARCH

The database that was used in this research was not explicitly captured in the context of a smart transportation system. The public domain WiAR dataset is used to detect human activities. The SVM classification method was applied in two sets of data, 1) activities with lower movement energy and 2) higher movement energy. Activities with

lower movement energy such as walking and standing are considered analogous to people waiting for a vehicle. Since 1) the speed of a moving vehicle is higher than the speed of the running activity and 2) the running activity has the highest speed and movement energy in comparison to other human activities, it is assumed analogous to people who get into a vehicle.

XI. CONCLUSION

The present paper proposed a data-driven and machine-learning-based methodology for automatic Human Activity Recognition (HAR) that is analogous to passengers getting into public transportation systems. Characterizing features using combined PCA and the decision tree methods, we conclude that the chirp rate explains the highest variability and produces the highest information gain. Moreover, measuring the SVM classification metrics confirms that using only chirp rate as a feature, we achieve $TP = 94\%$, $TN = 91\%$, precision $= 83\%$ and F1-score $= 87\%$. Considering delta band power as an additional feature adds more information and achieves a higher performance and scores to $TP = 97\%$, $TN = 95\%$, precision $= 100\%$, and F1-score $= 95\%$. Adding each principal component increases information and improves the performance scores. The Pearson correlation coefficient confirms that the information obtained with the frequency of movement feature is a subset of the information obtained with the chirp feature and it causes data to be redundant.

There is a trade-off between latency, accuracy, and feature selection. Thus, we recommended a feature selection based on observed variability, information gain, and correlation between features. The results show that by using PCA, we reduced the number of features from four to three. By applying the decision tree algorithm after PCA, the number of features needed for HAR was decreased to two, which allows for reducing the computational time and the complexity.

In future work, we could detect and count people who are crossing each other and who are becoming hidden by fat and tall ones to estimate the flow of traffic and to separate people who are moving from those who are outside a vehicle. Localization is required to extract information about recognizing people around a transportation station. An alert system is required to inform when the number of people at a station exceeds some safety threshold or when there is nobody at a station. Enhancing the proposed model to multi-person detection is another possible improvement. A fully automated and driverless transportation system such as the REM [66] could be considered for validation of the proposed passenger activity identification system in a real-world application scenario and for further exploration of its effectiveness.

REFERENCES

[1] C. Zhang, "Design and application of fog computing and Internet of Things service platform for smart city," *Future Gener. Comput. Syst.*, vol. 112, pp. 630–640, Nov. 2020.

[2] A. Kumari and S. Tanwar, "Secure data analytics for smart grid systems in a sustainable smart city: Challenges, solutions, and future directions," *Sustain. Comput. Informat. Syst.*, vol. 28, Dec. 2020, Art. no. 100427.

[3] S. Kaffash, A. T. Nguyen, and J. Zhu, "Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis," *Int. J. Prod. Econ.*, vol. 231, Jan. 2021, Art. no. 107868.

[4] D. Garcia-Retuerta, P. Chamoso, G. Hernández, A. S. R. Guzmán, T. Yigitcanlar, and J. M. Corchado, "An efficient management platform for developing smart cities: Solution for real-time and future crowd detection," *Electronics*, vol. 10, no. 7, p. 765, 2021.

[5] A. Novak, A. N. Sedlackova, M. Vochozka, and G. H. Popescu, "Big data-driven governance of smart sustainable intelligent transportation systems: Autonomous driving behaviors, predictive modeling techniques, and sensing and computing technologies," *Contempor. Read. Law Soc. Just.*, vol. 14, no. 2, pp. 100–117, 2022.

[6] D. Alahakoon, R. Nawaratne, Y. Xu, D. De Silva, U. Sivarajah, and B. Gupta, "Self-building artificial intelligence and machine learning to empower big data analytics in smart cities," *Inf. Syst. Front.*, vol. 25, pp. 221–240, Feb. 2023.

[7] D. Liu, Z. Cao, M. Hou, H. Rong, and H. Jiang, "Pushing the limits of transmission concurrency for low power wireless networks," *ACM Trans. Sens. Netw.*, vol. 16, no. 4, pp. 1–29, 2020.

[8] Q. Ma, Z. Cao, W. Gong, and X. Zheng, "BOND: Exploring hidden bottleneck nodes in large-scale wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 17, no. 2, pp. 1–21, 2021.

[9] (Otonomo, San Francisco, CA, USA). "Smart cities," Mar. 2023. [Online]. Available: https://otonomo.io/blog/smart-cities-intelligent-transportation-systems

[10] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1840–1852, Mar. 2021.

[11] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car detection using unmanned aerial vehicles: Comparison between faster R-CNN and YOLOv3," in *Proc. 1st Int. Conf. Unmanned Veh. Syst. Oman (UVS)*, 2019, pp. 1–6.

[12] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2016, pp. 3464–3468.

[13] X. Zheng et al., "Big data for social transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 620–630, Mar. 2016.

[14] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[15] A. I. Abubakar, K. G. Omeke, M. Ozturk, S. Hussain, and M. A. Imran, "The role of artificial intelligence driven 5G networks in COVID-19 outbreak: Opportunities, challenges, and future outlook," *Front. Commun. Netw.*, vol. 1, p. 4, Nov. 2020.

[16] C. Yin, J. Chen, X. Miao, H. Jiang, and D. Chen, "Device-free human activity recognition with low-resolution infrared array sensor using long short-term memory neural network," *Sensors*, vol. 21, no. 10, p. 3551, 2021.

[17] S. H. Khan, M. H. Yousaf, F. Murtaza, and S. Velastin, "Passenger detection and counting for public transport system.," *NED Univ. J. Res.*, vol. 17, no. 2, pp. 35–46, 2020.

[18] M. T. Nguyen, "Distributed compressive and collaborative sensing data collection in mobile sensor networks," *Internet Things*, vol. 9, Mar. 2020, Art. no. 100156.

[19] E. Hyun and Y. Jin, "Doppler-spectrum feature-based human–vehicle classification scheme using machine learning for an FMCW radar sensor," *Sensors*, vol. 20, no. 7, p. 2001, 2020.

[20] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 65–76.

[21] R. Alizadeh, Y. Savaria, and C. Nerguizian, "Human activity recognition and people count for a SMART public transportation system," in *Proc. IEEE 4th 5G World Forum (5GWF)*, 2021, pp. 182–187.

[22] C. Mateo and J. A. Talavera, "Short-time Fourier transform with the window size fixed in the frequency domain," *Digit. Signal Process.*, vol. 77, pp. 13–21, Jun. 2018.

[23] Wiki. "Chirp." Mar. 2023. [Online]. Available: https://en.wikipedia.org/wiki/Chirp,

[24] M. Barbu, E. J. Kaminsky, and R. E. Trahan, "Fractional fourier transform for sonar signal processing," in *Proc. OCEANS MTS/IEEE*, 2005, pp. 1630–1635.

[25] A. Gómez-Echavarría, J. P. Ugarte, and C. Tobón, "The fractional fourier transform as a biomedical signal and image processing tool: A review," *Biocybern. Biomed. Eng.*, vol. 40, no. 3, pp. 1081–1093, 2020.

[26] M. Soumekh, *Synthetic Aperture Radar Signal Processing*, vol. 7. New York, NY, USA: Wiley, 1999.

[27] K. Wang, X. Cheng, Y. Chen, C. She, K. Sun, and P. Zhao, "Heart sound model based on cascaded and lossless acoustic tubes," *J. Mech. Med. Biol.*, vol. 19, no. 5, 2019, Art. no. 1950031.

[28] J. Chen, H. Yu, G. Xu, J. Zhang, B. Liang, and D. Yang, "Airborne SAR autofocus based on blurry imagery classification," *Remote Sens.*, vol. 13, no. 19, p. 3872, 2021.

[29] G. Yu, J. Liang, W. Fan, H. C. So, and D. Zhou, "Autofocus algorithms with phase error correction for synthetic aperture radar imagery," *Digit. Signal Process.*, vol. 130, Oct. 2022, Art. no. 103692.

[30] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Data Sets*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[31] R. Alizadeh, Y. Savaria, and C. Nerguizian, "Automatic detection of people getting into a bus in a smart public transportation system," in *Proc. 29th IEEE Int. Conf. Electron. Circuits Syst. (ICECS)*, 2022, pp. 1–4.

[32] Computer Science. "Entropy." Feb. 2023. [Online]. Available: https://cs.stackexchange.com/questions/136106/what-is-the-relationship-between-entropy-and-variance

[33] O. Oshiga, H. U. Suleiman, S. Thomas, P. Nzerem, L. Farouk, and S. Adeshina, "Human detection for crowd count estimation using CSI of WiFi signals," in *Proc. 15th Int. Conf. Electron. Comput. Comput. (ICECCO)*, 2019, pp. 1–6.

[34] S. Liu, Y. Zhao, F. Xue, B. Chen, and X. Chen, "DeepCount: Crowd counting with WiFi via deep learning," 2019, *arXiv:1903.05316*.

[35] Y. Zhou, L. Liu, H. Zhao, M. López-Benítez, L. Yu, and Y. Yue, "Towards deep radar perception for autonomous driving: Datasets, methods, and challenges," *Sensors*, vol. 22, no. 11, p. 4208, 2022.

[36] S.-W. Kang, M.-H. Jang, and S. Lee, "Identification of human motion using radar sensor in an indoor environment," *Sensors*, vol. 21, no. 7, p. 2305, 2021.

[37] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM–A tutorial into long short-term memory recurrent neural networks," 2019, *arXiv:1909.09586*.

[38] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning*, Amsterdam, The Netherlands: Elsevier, 2020, pp. 101–121.

[39] X. Hu, Y. Che, X. Lin, and S. Onori, "Battery health prediction using fusion-based feature selection and machine learning," *IEEE Trans. Transport. Electrific.*, vol. 7, no. 2, pp. 382–398, Jun. 2021.

[40] R. Kalakoti, S. Nõmm, and H. Bahsi, "In-depth feature selection for the statistical machine learning-based Botnet detection in IoT networks," *IEEE Access*, vol. 10, pp. 94518–94535, 2022.

[41] S. Akhter, S. Ahmed, A. Kabir, and M. Shoyaib, "A relief based feature subset selection method," *Dhaka Univ. J. Appl. Sci. Eng.*, vol. 6, no. 2, pp. 7–13, 2021.

[42] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proc. 9th Int. Workshop Mach. Learn.*, 1992, pp. 249–256.

[43] B. M. S. Hasan and A. M. Abdulazeez, "A review of principal component analysis algorithm for dimensionality reduction," *J. Soft Comput. Data Min.*, vol. 2, no. 1, pp. 20–30, 2021.

[44] J. P. Bharadiya, "A tutorial on principal component analysis for dimensionality reduction in machine learning," *Int. J. Innovat. Sci. Res. Technol.*, vol. 8, no. 5, pp. 2028–2032, 2023.

[45] B. T. Jijo and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *J. Appl. Sci. Technol. Trends*, vol. 2, no. 1, pp. 20–28, 2021.

[46] Y. Yuan, L. Wu, and X. Zhang, "Gini-impurity index analysis," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3154–3169, 2021.

[47] J. Cui and D. Wang, "Biosignal analysis with matching-pursuit based adaptive chirplet transform," 2017, *arXiv:1709.08328*.

[48] Y. Lu, R. Demirli, G. Cardoso, and J. Saniie, "A successive parameter estimation algorithm for chirplet signal decomposition," *IEEE Trans. Ultrason., Ferroelect., Freq. Control*, vol. 53, no. 11, pp. 2121–2131, Nov. 2006.

[49] J. S. Lim and A. V. Oppenheim, *Advanced Topics in Signal Processing*. Hoboken, NJ, USA: Prentice-Hall, 1987.

[50] H. Chen, Z. Liu, B. Wu, and C. He, "A technique based on nonlinear hanning-windowed chirplet model and genetic algorithm for parameter estimation of lamb wave signals," *Ultrasonics*, vol. 111, Mar. 2021, Art. no. 106333.

[51] E. Priya, S. J. Priyadharshini, P. S. Reshma, and S. Sashaank, "Temporal and spectral features based gender recognition from audio signals," in *Proc. Int. Conf. Commun. Comput. Internet Things (IC3IoT)*, 2022, pp. 1–5.

[52] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *Proc. 13th {USENIX} Symp. Netw. Syst. Design Implement. ({NSDI})*, 2016, pp. 165–178.

[53] A. V. Oppenheim, A. S. Willsky, S. H. Nawab, and J.-J. Ding, *Signals and Systems*, vol. 2. Upper Saddle River, NJ, USA: Prentice Hall, 1997.

[54] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge, U.K.: Cambridge Univ. Press, 2019.

[55] J. Shlens, "A tutorial on principal component analysis," 2014, *arXiv:1404.1100*.

[56] Numpy. "Corrcoef." Jan. 2023. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.corrcoef.html

[57] B. Cael, K. Bisson, E. Boss, and Z. K. Erickson, "How many independent quantities can be extracted from ocean color?" *Limnol. Oceanogr. Lett.*, vol. 8, no. 4, pp. 603–610, 2023.

[58] WiKi. "Entropy." Sep. 2023. [Online]. Available: https://en.wikipedia.org/wiki/Entropy_(information_theory)

[59] WiKi. "Variance." Sep. 2023. [Online]. Available: https://en.wikipedia.org/wiki/Variance

[60] B. T. Jijo and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Evaluation*, vol. 6, no. 1, p. 7, 2021.

[61] J. Liu, C. Wang, W. Chi, G. Chen, and L. Sun, "Estimated path information gain-based robot exploration under perceptual uncertainty," *Robotica*, vol. 40, no. 8, pp. 2748–2764, 2022.

[62] Wiki. "Scree plot." Feb. 2023. [Online]. Available: https://en.wikipedia.org/wiki/Scree_plot

[63] S. Raschka, Y. H. Liu, V. Mirjalili, and D. Dzhulgakov, *Machine Learning With PyTorch and Scikit-Learn: Develop Machine Learning and Deep Learning Models With Python*. Birmingham, U.K.: Packt Publ. Ltd., 2022.

[64] Scikit-learn. "Decision-tree." Apr. 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot_tree.html

[65] A. Tharwat, "Classification assessment methods," *Appl. Comput. Informat.*, vol. 17, no. 1, pp. 168–192, 2021.

[66] WiKi. "REM." Mar. 2023. [Online]. Available: rem.info/en/light-rail